UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
FACULTAD DE CONTADURÍA Y ADMINISTRACIÓN

ALGORITMOS GENÉTICOS

TEMAS SELECTOS DE ESTADÍSTICA
LIC. EN ACTUARÍA
SÉPTIMO SEMESTRE

MAURICIO ALMARAZ GONZALEZ
HUGO ALEJANDRO RESÉNDIZ NAVA

Optimización del Problema del Viajero con Algoritmos Genéticos

Los algoritmos genéticos son técnicas inspiradas en la evolución biológica, utilizadas para resolver problemas de optimización complejos, como el **Traveling Salesman Problem** (TSP). En este proyecto, se desarrolló una solución en Python utilizando dos enfoques: uno basado en visualizaciones con `matplotlib` y otro con visualización interactiva mediante `pygame`.

Conceptos Clave

- **Población Inicial**: Se generan rutas aleatorias que comienzan y terminan en la ciudad 'A'.
- **Genes**: Las rutas representan secuencias de ciudades a visitar, lo que define la solución del problema.
- **Fitness**: Se evalúa según la distancia total recorrida. Cuanto menor sea la distancia, mejor será la solución.
- **Selección**: Se seleccionan las mejores rutas ordenadas por distancia para reproducirse.
- **Crossover**: Combina segmentos de dos rutas para crear una nueva, preservando parte del orden original.
- **Mutación**: Intercambia posiciones de dos ciudades en una ruta con cierta probabilidad.

Proceso de Evolución

1. **Generación Inicial**: Se crean rutas aleatorias.
2. **Evaluación y Selección**: Se calculan distancias y se eligen las mejores rutas.
3. **Reproducción**: Se generan nuevas rutas mediante *crossover* y mutación.
4. **Iteración**: El proceso se repite hasta alcanzar un número definido de generaciones o una solución óptima.

Mejoras Futuras

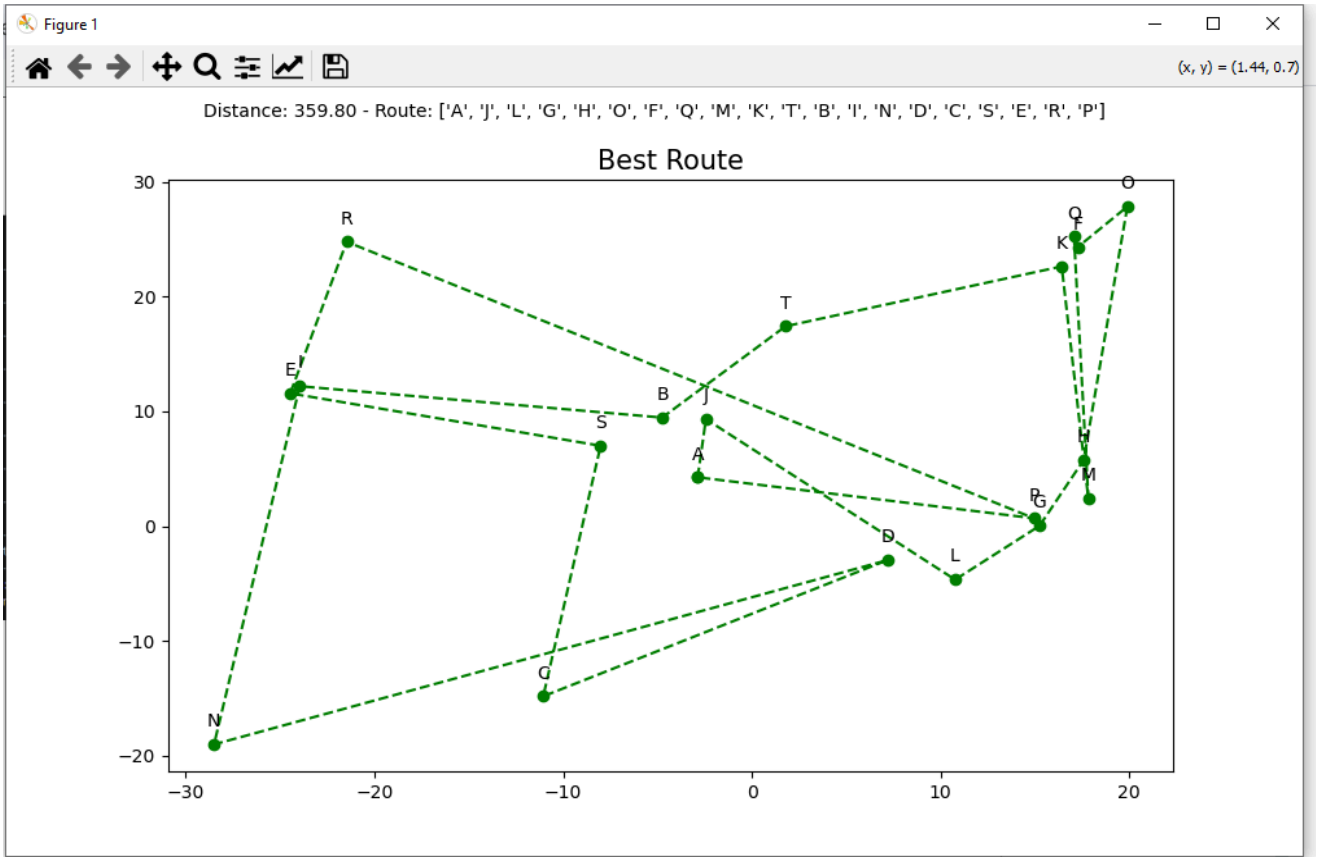
- Ajustar el sistema de selección para balancear distancia y número de pasos.
- Implementar rutas con movimientos más complejos, no limitados a coordenadas fijas.

Ejecución Visual

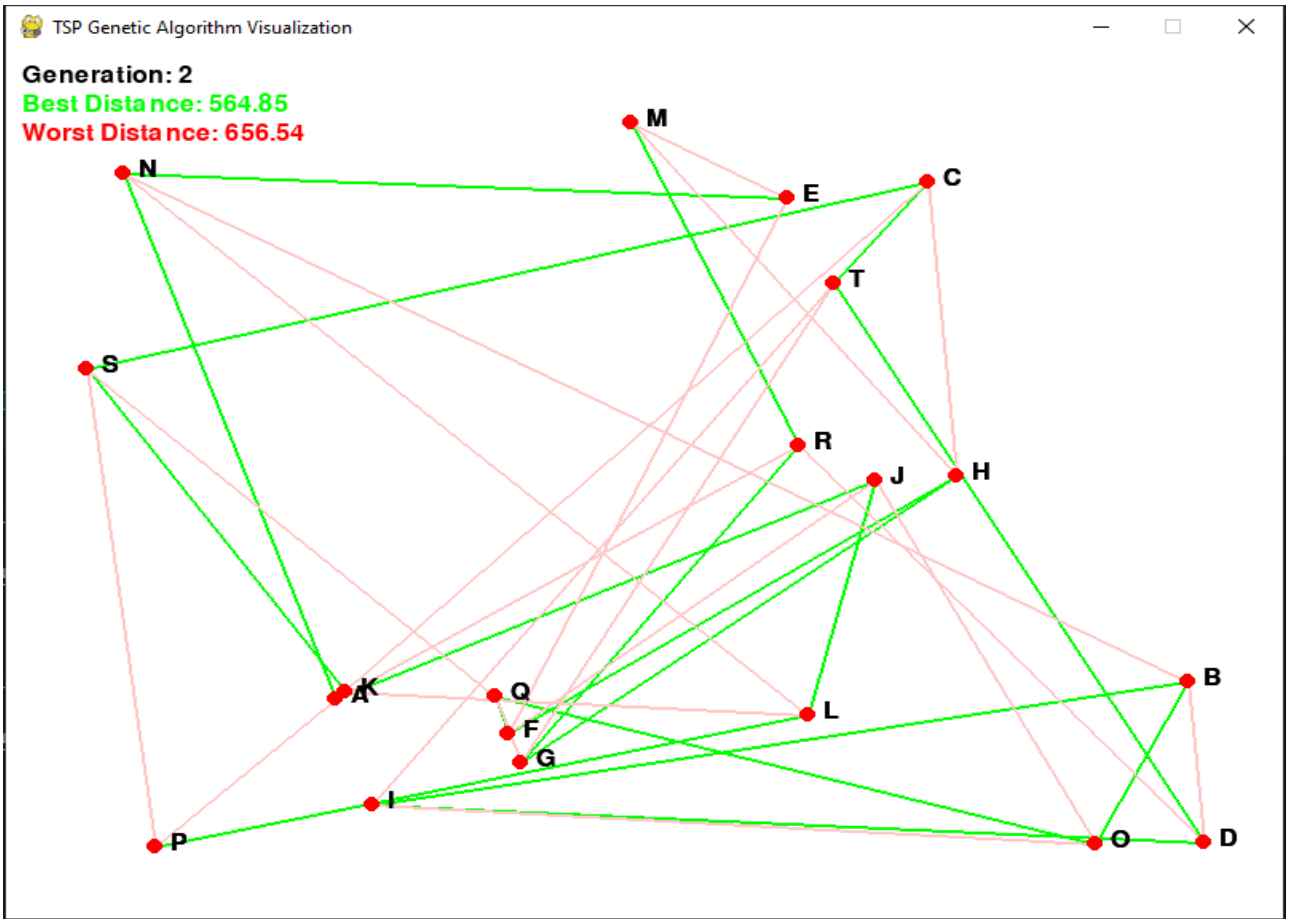
El segundo archivo incluye una representación visual del proceso utilizando `pygame`, lo que permite observar el progreso de las rutas óptimas y peores en tiempo real, facilitando el análisis dinámico de la evolución del algoritmo.

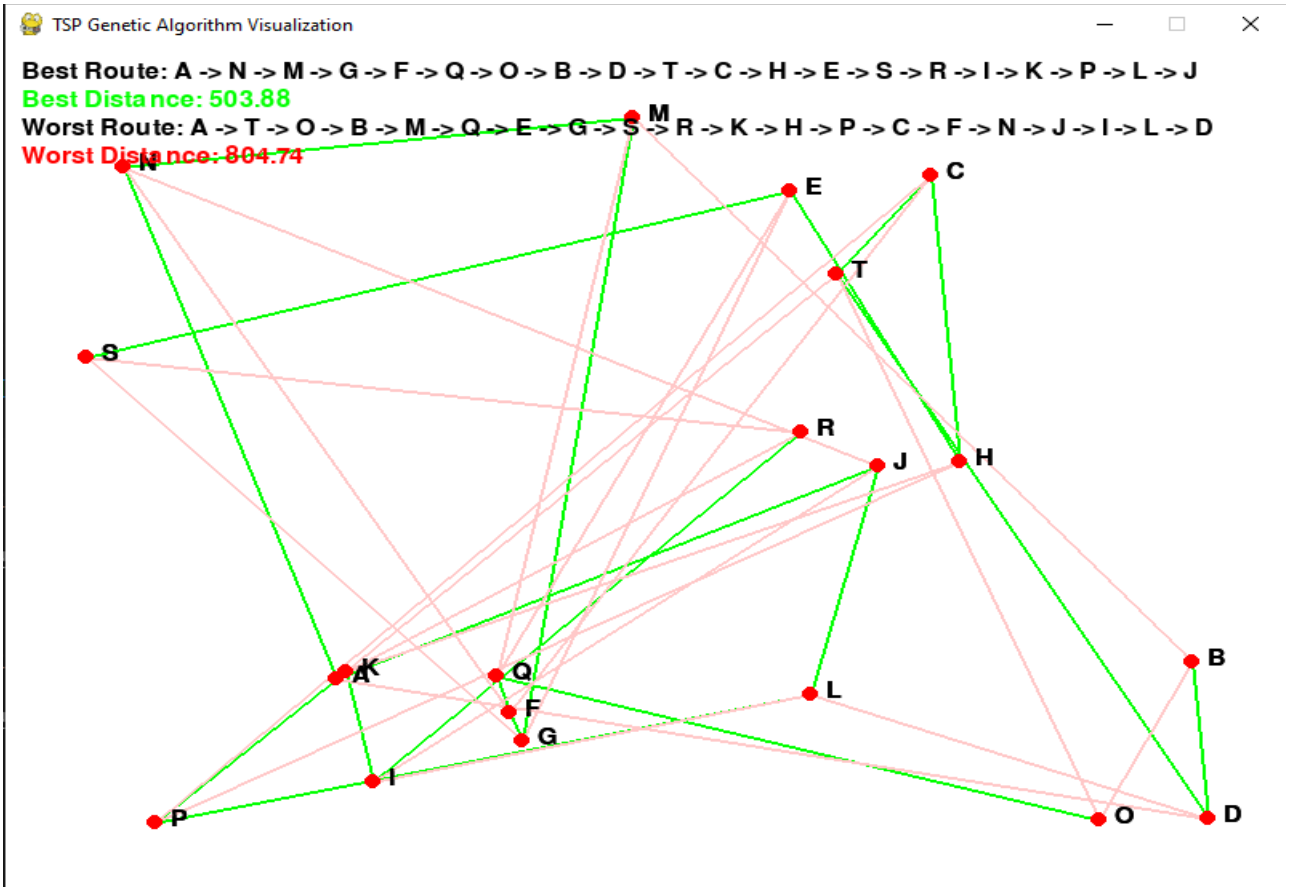
main.py

```
PS C:\Users\maugo> & C:/Users/maugo/AppData/Local/anaconda3/python.exe c:/Users/maugo/Documents/Proyecto_Final/main.py
Generation 1: Best Route: ['A', 'H', 'D', 'G', 'C', 'L', 'R', 'Q', 'F', 'M', 'J', 'K', 'O', 'T', 'N', 'I', 'S', 'P', 'E', 'B'] -> Distance: 459.43
--------------------------------------------------
Generation 2: Best Route: ['A', 'H', 'D', 'G', 'C', 'L', 'R', 'Q', 'F', 'M', 'J', 'K', 'O', 'T', 'N', 'I', 'S', 'P', 'E', 'B'] -> Distance: 459.43
--------------------------------------------------
Generation 3: Best Route: ['A', 'I', 'N', 'S', 'J', 'L', 'G', 'K', 'T', 'E', 'H', 'O', 'F', 'B', 'D', 'C', 'R', 'Q', 'M', 'P'] -> Distance: 442.45
--------------------------------------------------
Generation 4: Best Route: ['A', 'J', 'L', 'G', 'H', 'O', 'F', 'B', 'Q', 'M', 'C', 'D', 'K', 'T', 'N', 'I', 'S', 'P', 'E', 'R'] -> Distance: 439.73
--------------------------------------------------
Generation 5: Best Route: ['A', 'J', 'L', 'G', 'H', 'O', 'F', 'B', 'Q', 'M', 'C', 'D', 'K', 'T', 'N', 'I', 'S', 'E', 'R', 'P'] -> Distance: 426.24
--------------------------------------------------
Generation 6: Best Route: ['A', 'J', 'L', 'G', 'H', 'O', 'F', 'B', 'I', 'N', 'S', 'K', 'T', 'E', 'D', 'C', 'R', 'Q', 'M', 'P'] -> Distance: 426.09
--------------------------------------------------
Generation 7: Best Route: ['A', 'J', 'L', 'G', 'H', 'O', 'F', 'B', 'I', 'N', 'S', 'K', 'T', 'E', 'D', 'C', 'R', 'Q', 'M', 'P'] -> Distance: 426.09
--------------------------------------------------
Generation 8: Best Route: ['A', 'J', 'L', 'G', 'H', 'O', 'F', 'Q', 'M', 'K', 'T', 'N', 'I', 'B', 'D', 'C', 'S', 'E', 'R', 'P'] -> Distance: 374.99
--------------------------------------------------
Generation 9: Best Route: ['A', 'J', 'L', 'G', 'H', 'O', 'F', 'Q', 'M', 'K', 'T', 'N', 'I', 'B', 'D', 'C', 'S', 'E', 'R', 'P'] -> Distance: 374.99
--------------------------------------------------
Generation 10: Best Route: ['A', 'J', 'L', 'G', 'H', 'O', 'F', 'Q', 'M', 'K', 'T', 'B', 'I', 'N', 'D', 'C', 'S', 'E', 'R', 'P'] -> Distance: 359.80
--------------------------------------------------
Best overall route: ['A', 'J', 'L', 'G', 'H', 'O', 'F', 'Q', 'M', 'K', 'T', 'B', 'I', 'N', 'D', 'C', 'S', 'E', 'R', 'P']
Best overall distance: 359.80
```



main2.py


TSP Genetic Algorithm Visualization
Generation: 2
Best Distance: 564.85
Worst Distance: 656.54
M
N
E
C
T
S
R
J
H
K
A
Q
B
L
F
G
I
P
O
D


TSP Genetic Algorithm Visualization
Best Route: A -> N -> M -> G -> F -> Q -> O -> B -> D -> T -> C -> H -> E -> S -> R -> I -> K -> P -> L -> J
Best Distance: 503.88
Worst Route: A -> T -> O -> B -> M -> Q -> E -> G -> S -> R -> K -> H -> P -> C -> F -> N -> J -> I -> L -> D
Worst Distance: 804.74
M
N
E
C
T
S
R
J
H
K
A
Q
B
L
F
G
I
P
O
D